# ;login:
The USENIX Association Newsletter

Volume 9 Number 5  
November 1984

## CONTENTS

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Winter 1985 USENIX Conference</td>
<td>3</td>
</tr>
<tr>
<td>Tutorials and Technical Program</td>
<td>3</td>
</tr>
<tr>
<td>Introduction</td>
<td>5</td>
</tr>
<tr>
<td>Narration</td>
<td>5</td>
</tr>
<tr>
<td>Where We Are</td>
<td>9</td>
</tr>
<tr>
<td>European UNIX System User Group Meeting</td>
<td>10</td>
</tr>
<tr>
<td>USENIX Conference Proceedings Available</td>
<td>30</td>
</tr>
<tr>
<td>The First BiMonthly UNIX Crossword Puzzle</td>
<td>31</td>
</tr>
<tr>
<td>Local User Groups</td>
<td>32</td>
</tr>
<tr>
<td>USENIX Searching For An Executive Director</td>
<td>33</td>
</tr>
<tr>
<td>4.2BSD Buglist To Be Available As 84.2 USENIX Distribution Tape</td>
<td>33</td>
</tr>
<tr>
<td>1984 Software Distribution Tape Submissions</td>
<td>34</td>
</tr>
<tr>
<td>Contents of the 84.1 USENIX Tape Submissions</td>
<td>34</td>
</tr>
<tr>
<td>Release Form for 1985 USENIX Distribution Tapes</td>
<td>36</td>
</tr>
<tr>
<td>Application for Membership for 1985</td>
<td>39</td>
</tr>
<tr>
<td>Announcing Second Printing of USENIX 4.2BSD Manuals</td>
<td>41</td>
</tr>
<tr>
<td>4.2BSD Manual Reproduction Authorization and Order Form</td>
<td>43</td>
</tr>
</tbody>
</table>
NOTICE

;login: is the official newsletter of the USENIX Association, and is sent free of charge to all members of the Association.

The USENIX Association is an organization of AT&T licensees, sub-licensees, and other persons formed for the purpose of exchanging information and ideas about UNIX\(^1\) and UNIX-like operating systems and the C programming language. It is a non-profit corporation incorporated under the laws of the State of Delaware. The officers of the Association are:

<table>
<thead>
<tr>
<th>Office</th>
<th>Name</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>President</td>
<td>Alan G. Nemeth</td>
<td>Directors</td>
</tr>
<tr>
<td>Vice-President</td>
<td>Deborah K. Scherrer</td>
<td>Thomas Ferrin</td>
</tr>
<tr>
<td>Secretary</td>
<td>Lewis A. Law</td>
<td>Steve C. Johnson</td>
</tr>
<tr>
<td>Treasurer</td>
<td>Waldo M. Wedel</td>
<td>Lou Katz</td>
</tr>
</tbody>
</table>

The editorial staff of ;login: is:

<table>
<thead>
<tr>
<th>Role</th>
<th>Name</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technical Editor</td>
<td>Brian Redman</td>
<td>ucbvax</td>
</tr>
<tr>
<td>Copy Editor</td>
<td>Michelle Peetz</td>
<td>ucbvax</td>
</tr>
<tr>
<td>Managing Editor</td>
<td>Tom Strong</td>
<td>ucbvax</td>
</tr>
<tr>
<td>Editorial Director</td>
<td>Lou Katz</td>
<td>ucbvax</td>
</tr>
</tbody>
</table>

Member services are provided through the Association office. Membership information can be obtained from the office:

USENIX Association
P.O. Box 7
El Cerrito, CA 94530
(415) 528-8649

[ucbvax|decvax|luseunix|office]

Members of the UNIX community are heartily encouraged to contribute articles and suggestions for ;login:. Your contributions may be sent to the editors electronically at the addresses above or through the U.S. mail to the Association office at the address above. The USENIX Association reserves the right to edit submitted material.

;login: is produced on UNIX using troff and a variation of the —me macros. We appreciate receiving your contributions in n/troff input format, using any macro package. If you contribute hardcopy articles please leave left and right margins of 1" and a top margin of 1 1/2" and a bottom margin of 1 1/4". Hardcopy output from a line printer or most dot-matrix printers is not reproducible.

Acknowledgments

The Association uses a VAX\(^4\) 11/730 donated by the Digital Equipment Corporation for support of office and membership functions, preparation of ;login:, and other Association activities. It runs 4.2BSD, which was contributed and installed and is maintained by Mt Xinu. The VAX uses a sixteen line VMZ-32 terminal multiplexor donated by Able Computer of Irvine, California.

This newsletter may contain information covered by one or more licenses, copyrights, and/or non-disclosure agreements. No reproduction of this newsletter in its entirety or in part may be made without written permission of the Association.

\(^1\)UNIX is a trademark of Bell Laboratories. \(^4\)VAX is a trademark of Digital Equipment Corporation.
Winter 1985 USENIX Conference

The Winter 1985 USENIX Conference is scheduled for January 23–25, 1985, at the Fairmont Hotel in Dallas, Texas. The conference will feature eight tutorials, two days of technical presentations, Birds of a Feather sessions on two evenings, and a meeting of the USENIX Board of Directors with the members; the schedule for these events is given below. The conference is being held concurrently with /usr/group’s UniForum Tradeshow, which will be at the Infomart in Dallas on January 21–25.

USENIX Conference Proceedings, containing all papers submitted prior to the conference, will be distributed at the conference. Additional copies of the Proceedings may be purchased at the registration deck or may be ordered after the conference from the USENIX Office.

A registration packet was mailed to a large mailing list on November 23 and 24. It contains details on the tutorials and fees, a registration form, a hotel reservation form, information on special airfares, etc. If you did not receive a registration packet please contact

USENIX Conference Office (213) 592-1381 or (213) 592-3243
P.O. Box 385
Sunset Beach, CA 90742

The deadline for hotel reservations is December 23 and for conference pre-registration is January 7 (after that the registration and tutorial fees go up).

Tutorials and Technical Program

A set of eight tutorials will be held concurrently on Wednesday, January 23, from 9am until 5pm. They have been organized by Michael Tilson of Human Computing Resources Corporation. The tutorials will be intense, in-depth sessions, with attendance limited to approximately 150 persons each. Pre-registration is strongly advised; on-site registration will be allowed, if space permits. Attendees at the 4.2BSD Internals tutorial must be covered under a 4.2BSD source license; information on license verification is contained in the registration packet.

Charisse Castagnoli of Teknekon Infoswitch Corp. and Rob Kolstad of Convex Computer Corp. are the Conference Program Co-Chairs. They have now received dozens of abstracts of various papers for the technical program. Of these, most have been accepted for presentation (conditional upon receipt of papers of corresponding quality). After refereeing by the Conference and session chairs, all accepted papers will be published in the proceedings. This will allow the technical aspect of the conference to be of the highest possible caliber for a gathering of this type.

Birds of a Feather sessions (BOFs) may be scheduled in advance by contacting the USENIX Conference Office (see above). BOFs may, of course, also be scheduled on-site.

WEDNESDAY, JANUARY 23
9:00am - 5:00pm  Tutorials
4.2BSD Internals (4.2BSD source license required)
   Kirk McKusick and Mike Karels, U.C. Berkeley
The ANSI C Standard
   Larry Rosler, AT&T Bell Laboratories
UNIX Networking
   Bruce Borden, Silicon Graphics, Inc.
Advanced C Programming, Illustrated by Classical Algorithms
   Walter Brown, Moravian College
Writing UNIX Device Drivers
   Ken Greer, Elan Computer Group
UNIX Language Tools
   William Appelbe, U.C. San Diego
UNIX Interprocess Communication
   Robert Hutchison, Auxco
   uucp, Mail, and News
   Mark Stein, Fortune Systems
UNIX System Administration
   Ed Gould, mt Xinu

Evening                                                                 Birds of a Feather Sessions

THURSDAY, JANUARY 24
9:00am - 9:15am  Opening Remarks
9:15am - 10:00am Keynote Speaker
10:30am - 12:30pm Languages
   Chair: Steve Johnson, AT&T Bell Laboratories
   Topics: Implementations of Various Languages on UNIX,
   Panel Discussion of C versus Other Languages on UNIX
1:30pm - 3:00pm  Kernel Implementation
   Chair: John Quarterman, University of Texas, Austin
   Topics: Recent Implementations of UNIX
3:30pm - 5:00pm  User Interfaces
   Chair: Alfred Corrieria, Computer Thought Corp.
   Topics: New UNIX Front Ends; Windowing
5:30pm - 7:00pm  Meeting of the Board of Directors with the Members
Evening                                                                 Birds of a Feather Sessions
   Films: A Guided Tour of Program Design Methodologies and Software Quality

FRIDAY, JANUARY 25
9:00am - 10:00am uucp & Usenet
   Chair: Tom Watson, Scientific Machines Corp.
   Topics: Current Status of Usenet Mapping Projects
10:30am - 11:30am Networking
   Chair: Joe Kalash, University of California, Berkeley
   Topics: Kernel Networking Support
11:30am - 12:30pm Standards/Directions
   Chair: John Chambers, Microelectronics Center
   Topics: C++, Networking, Graphics, Portability Issues
1:30pm - 3:00pm  Software Tools/Applications
   Chair: John Trudeau, Teknekron Infoswitch
   Topics: SETOPT — Command Line Option Parser Generator,
   Productivity Tools and Performance Issues
3:30pm - 5:00pm  Real Time/Distributed Systems
   Chair: Tom Ferrin, University of California, San Francisco
   Topics: Device Drivers in a Multiprocessor Environment

TO BE SCHEDULED

Netnews via Satellite: A Progress Report
   Lauren Weinstein, Vortex Technology
Research into Liability Issues in Netnews Transmission
   Susan Nycum, Attorney, Gaston Snow & Ely Bartlett

November 1984  Volume 9, Number 5
After Newcastle, What?

A Report on the Distributed UNIX Meeting
Newport, RI, September 1984
Veigh S. Meer
Bell Communications Research
Morristown, NJ 07960

ABSTRACT

Consider the Cartesian product of: (kernel/library/shell) × (remote files/remote procedures) × (heterogeneous/homogeneous hardware) × (use what’s there/redo everything) × (works now/being built). That covers this conference pretty well: how to handle a larger user community by piecing together conventional UNIX service on smallish machines. The easiest thing to do is remote files via a new C library. It’s about time somebody started thinking about doing something new. When Multics was as old as UNIX, it had been dead for ten years.

Introduction

About 80 UNIX hackers met at the Viking Hotel, Newport, RI Sept. 12-14, 1984 to talk about distributed UNIX systems. Among major projects represented were 4.2BSD (Leffler and McKusick), Newcastle (Lindsay Marshall), Athena (Dave Mankins & others), BB&N (Gurwitz), and SUN (Bob Lyons). Nobody spoke from CMU-ITC or Bell Laboratories research. This is an opinionated report which reflects only the views of the author.

This conference description will be divided into two parts. The first is simply a chronological narration of the talks, since there is no printed proceedings. This is followed by a summary of the state of the art, combined with my personal view of the future.

Narration

Doug Comer, from Purdue, described their TILDE project, a broadcast net allowing terminals to broadcast service requests (for whole processes) which servers then bid for. They imagine a basically conventional environment in which the distribution serves for load-balancing, thinking in terms of user files migrating around with a time constant in days, and for increasing community size. They have not faced up to authentication and dissimilar hardware, preferring to do something useful now. TILDE has similarities to a lot of dumb terminals connecting to standard UNIX hosts, with a self-adapting switch in between that evens out the user distribution on machines without a system administrator intervening.

Nancy Hall, of Wisconsin, followed with a talk on Crystal. A PRONET ring (as is TILDE), Crystal has an unusually large number of diskless CPU nodes (40 11/750s, of which 4 have disks). The idea is that a few machines (mostly 3 780s) support remote files for a lot of processes running on the CPUs. The operating system is Charlotte, not UNIX, but has similar properties and goals. Basically this is an attempt to increase the throughput of the three 780s by offloading CPU work to the 750s. It seemed the 750s had arrived for unspecified other reasons and Crystal was an attempt to make use of them. I think you can get the same benefits more cheaply by offloading to 68K boards (which can just be dedicated to particular host CPUs since they’re so cheap) and the architecture here has far too little disk and far too much CPU for typical use. Maybe they have lots of number-crunchers at Wisconsin.

Keith Lantz, of Ridge and Stanford, followed with a description of the Ridge operating system (ROS), based on the Stanford V-system. He was the purest of the pure, planning network-wide virtual
memory, global userids and authentication, etc. implemented on a message-based operating system. His demand for no execution cost in using the network prompted an interesting discussion among Leffler, Gurwitz, Marshall and Lantz about efficiencies of implementations; the general word was you have to put it in the kernel to make it fast. Lantz’s goals are so large (absolute network transparency, IPC with no programming cost, etc.) that neither 4.2BSD nor LOCUS is distributed by his standards. More seriously, I think his goals demand centralized administration, and I see a lot of pressure for distributed systems that do NOT have central control. We got UNIX partly because people did not want to submit to big computer center bureaucracies when computers got cheap enough for each group to have one; now I see the same phenomenon with workstations. Anyway the goal of ROS is one big operating system using message-passing down below so the parts can be scattered around. Entire campuses or even more can be a single computer system. If it works, the Justice Department will get them.

Joe Requa, of Livermore, next showed that the rich are like the rest of us, except they have to connect Crays while we connect Suns. They have 5 Crays (4 Cray-I, 1 Cray-XMP), 3 7600s, and only 5 VAXes. Two NSC Hyperchannels connect everything. Most of the talk was at the low level of what system primitives would be provided; they are providing a complex protocol for message handling which is intended to support UNIX tasks. This is hard because UNIX processes are expensive, none of the standard device drivers are appropriate for their network, and they have heterogeneous hardware. Livermore is intending to run full ISO; maybe you need Crays to keep up with that. What with ISO, datagrams, machine-independent reliable flow control, etc. this project may pour more cycles into networking than anybody else. It does not help that they are unwilling to change UNIX because they want to buy only binary licenses; Rob Kolstad was among many who marveled at the thought of buying more Crays than VAXes but saving on software licenses.

John Forecast, of DEC, described the job of making 4.2BSD talk to DECNET. His talk convinced everybody that it was a bad idea. DECNET is large and has many complex options (e.g. optional connections) not known to UNIX. Making the link without changing DECNET is both complex and limiting. However, the job was done in impressively little time (6 man months, no previous UNIX experience).

Peter Reiher, of UCLA, next described how Locus will keep track of names. This was a fairly routine description of a plan for name servers keeping track of mounted file systems, each server being very reliable, and each workstation caching its local needs.

Dave Mankins, from MIT, now talked about naming in Athena. There was a long discussion about their convention (/@name for a remote name) which seems to get people overly excited. More interesting is that their file opens return connections, not files, and can actually be pipe-type connections to processes. Athena achieves 30Kbytes/second transfer rate over the network, and can also do remote procedure call. The basic protocol is datagrams. Athena was unusual at this meeting for its scale, and seemed fairly far along towards accomplishment.

There followed the most outspoken speaker of the meeting. Lindsay Marshall talked on the Newcastle connection. In two months he replaced the C library with remote file system routines, running on the Cambridge ring. They use / to signal remote names. His message was that remote file systems are easy. He did not claim it was exciting or new; but Newcastle publicized it and sold it. Personally, the earliest library rewrite I know of is Priscilla Lu’s R-library at Bell Labs in 1976; anyone know an earlier remote library? Marshall is now working on a triply redundant file system with majority voting for ultra-reliability.

Bill Appelbe of UCSD described the reverse of the Livermore situation: teaching 500 students Pascal on 55 IBM PCs. Each station has only a floppy, and the Omninet is slow. On the other hand, all the students have to run is Pascal, and it is more important to restrict them (for security and resource conservation) than to get work done. Most of the discussion was security; students can not even unmount their floppy's without logging out. But the equipment limits here were so serious that much of the design is seriously distorted from the viewpoint of a programmer or researcher.

Chris Kent at Purdue described a disk block cache. Disk caching is harder than file caching because of locking problems; the talk and discussion covered write-through, multiple caches, and
similar techniques. He proposed a broadcast protocol. This work is just beginning. Bob Lenk (HP) suggested reasonably enough that simultaneous writing was so rare that canceling caching in such cases would cause no overall performance penalty.

Dan Franklin from Interactive was also doing disk block caching. Again, the goal is remote file access with full support of upward movement (*chdir* ...), many networks, and kernel processes. Franklin has protocol details worked out to handle write-through and avoid races.

The next few talks were short talks given in an extension session; being shorter, they get shorter summaries.

Mike Dean, of BB&N, discussed the Cronus system, 9 hosts on an Ethernet providing an object-based operating system on top of some host. This project conforms to standard goals (remote file system, remote procedure call, reliability, authentication, message-based) but my bet is that the combination of flexibility, data abstraction, and user-space code will make for poor performance.

Dave Korn of AT&T-Bell Labs had put UNIX on top of Apollo Domain. He added //x to ... and /@x as names for the global file system.

Wayne Hathaway of NASA-Ames envisages an even larger and more expensive world than Livermore, as part of a national resource for computational fluid dynamics. Think about a Cray-2 with 500 MB of main memory, 200 GB of mass storage, 25 Silicon Graphics IRIS workstations as terminals, and Hyperchannels connecting them. Then go back to your ADM3a and cry. (Or think about swapping 500 MB or backing up 200 GB and smirk.) Anyway this was not a distributed operating system, or even a distributed file system. The users have to know where everything is. This project was an example of “pride in plagiarism” (speaker’s words): they took lots of System V, 4.2BSD sockets, Newcastle, etc. Or maybe it was mostly pride in hardware.

There followed the fastest contrast in the meeting: Larry Phillips of the University of Toronto is trying to teach 6000 engineers on 2 VAXes. They use NSC 32016 microprocessors (previously named 16032) on a Hubnet and limit the VAXes to fileservers. Quick description of the software is Newcastle in the kernel.

Divyakant Agrawal of SUNY Stony Brook is building a name server for a distributed relational data base system. A quick description of this software was Newcastle for relations.

Now we return to the full-length talks. Mike Lesk of Bell Communications Research spoke next, arguing that multi-processors should make one job faster, not just run more jobs. Furthermore he does not like CSP-type languages for programming. So he proposed rewriting troff in a concurrent production systems language, e.g. OPS-V. This permits dividing either the workspace, or the rule space, or doing multiple productions at once. The audience was generally skeptical. Martin Levy has distributed make, if you want a particular program that works now. This talk, like many others, was premature (not enough work yet).

Bob Lyon, of SUN, described the SUN language (XDR) for representing data on a network. XDR is supposed to deal with byte order problems, pointers, etc. Each data type must come with a routine to describe it. SUN has written routines for typical C types; plus the IEEE floating point standard is used. Some suggested there might be international standards coming, but I doubt it. An important data type missing, according to Dave Korn, was the bitmap. The cost of remote procedure call between dissimilar machines with binary arguments looks like it will be high, no matter what techniques are used.

Mike Hawley, of Lucasfilm, compared the SUN and Blit user interfaces for graphic programming. He is building an audio library for sound effects (laser guns, stomach punches, etc.) and has programmed extensively on both the SUN and Blit. Hawley strongly preferred the Blit, saying Rob Pike got it right (e.g. a rectangle is 2 points, not 4 ints), and ported the Blit stuff to the SUN in a month. Sam Leffler had tried a sample job twice and it took 1/2 the code in the Blit version. This was the only evaluated comparison in the entire conference.
Jeff Langer, of AT&T Bell Labs, described another remote procedure call system based on messages. They provide the messages via virtual circuits, however, separating the RPC from the network protocol; they intend to run on any of X.25, Ethernet, RS232, etc. The RPC itself is fast (a few hundred microseconds per transaction) but the underlying network they are now using is expensive. Again, multiple layers to separate protocols neatly costs performance. The overall design of the message-based RPC resembled DMOS. This talk, and the next, both reference the Nelson/Birrell PARC work.

Bob Souza, of Athena, described their remote procedure call. It is intended to have 2000-3000 stations across the MIT campus, as part of a curriculum development experiment. They support both blocking and non-blocking calls, with non-blocking calls intended for operations such as writing on a printer when you are never really sure the characters got onto the paper anyway. The RPC is multi-language; Athena supports C, Lisp, Fortran and (possibly) Pascal. A datagram protocol based on UDP is used and arguments are passed by value. There was no performance data yet.

Jehan-Francois Paris, of UCSD, described a proposed fault-tolerant file system, based on multiple hosts. Replication is a file system attribute; consistency is checked on open and majority vote decides discrepancies. This was implemented in the user level library (i.e. replicated Newcastle).

Tim Hoel described UNIX on the Cray-II, a new machine "so fast that it can run an infinite loop in 6 seconds." They are porting System V, and trying to maintain very fast file transfers; this includes track I/O, non-blocking read and write, and enormous files, covering several 600 MB drives. Even though the machine costs $15M, the average customer spends more on disk than on CPU. They do file striping (putting consecutive tracks of a file on different disks for speed), use sector or track allocation depending on file size. Little of this talk dealt with distribution; too much of the audience seemed to think that multiple machines was what you did when you could not afford one machine big enough to handle all your customers.

Neil Wiedenhoffer, of Denelcor, described UNIX on the HEP. The HEP is a multiprocessor machine that can run up to 16 processors and up to 64 interleaved instruction streams ("we can't run an infinite loop as fast as the Cray, but we can run 50 of them at once"). The basic primitive will be fork() + create new stream. Each instruction stream is a 2-3 Mips stream. The software is based on System III, plus the Bourne shell and the 4.2BSD file system. The LAN is a switch. The software has some basic primitives for synchronizing by "wait for write" type operations. Debugging is difficult; they are working on a new debugger named "The Force" (described as "like duct tape, it has a dark side and a light side and it holds the universe together").

Finally, there was a summary session, featuring Lindsay Marshall, Doug Comer, Al Nemeth and Rob Gurwitz. It gave a good perspective on the meeting; a little coverage of what has been done followed by a lot of discussion of what needed to be done. UNIX has been made to work on distributed machines, but functionality has not been improved, Marshall pointed out. Jim Gettys emphasized that distribution was about control of resources; you have to achieve both local service and access to a community to get happy users of distributed systems. Nobody knows how to debug distributed systems, and there is little progress so far on fault tolerance. Nemeth wanted more information on performance; speed is needed. The summary session sort of tailed off, with discussions of control of resources and administration, but reaching no real conclusion. So far we really do not know much about doing authentication, connecting dissimilar CPUs, controlling and allocating resources, and how to do all of this while leaving enough cycles for the users to accomplish something. There's lots of opportunity left in the field.

The conference organizers should be complemented on the attendance. The audience was vocal and knowledgeable, and nobody was in a three-piece suit. Publicity exclusively by netnews and invitation seems to work. Discussions were very technical and productive; almost as much fun as hacking. One sour note for future planners to remember: although Newport was conveniently located and provided good weather, the Viking Hotel provided a noisy conference room, appeared to be built of parts that flunked the Best Western acceptance test, and charged at least one attendee who made reservations separately substantially LESS than the "special conference rate."
Where We Are

Those interested in this conference should also be aware of the CMU ITC project. It involves a plan for a standard workstation with a campus-wide Ethernet and cached files. Dave Rosenthal would be a good contact. Meanwhile, at Bell Labs Dennis Ritchie has replaced all the character handling and communications routines in the kernel, and on top of that Peter Weinberger has built a network-independent remote file system, using the Datakit switch as the first exemplar. Remote files are implemented in the kernel; you always execute on the original machine, and ordinary file system name syntax (e.g. /n/machinename/...) is used to name files.

Anyway, what of the actual conference? It is clearly possible to build distributed systems by changing any of the library, the kernel, or the shell. The library is smallest and easiest, hence Newcastle. This gives you the ability to access remote files. Next easiest is remote processes, but the UNIX processes are so expensive that this is not a good way to break up big jobs. Remote procedure call is now achievable on identical machines, but is still plagued with problems of data description on heterogeneous hardware.

In all cases, performance is a problem. The standard solution seems to be code in the kernel. What happens when everything is in the kernel I do not know; maybe we should go back to a RISC-type minimal kernel and shell, giving faster processes and command execution, and reducing the need to do kernel code.

There is a basic philosophical difference between combining existing systems, retaining local autonomy, and designing a grand university-wide plan. A clear contrast, for example, is between MIT-Athena and CMU-ITC. University politics, funding, and so forth may decide which road is taken; but there are many hard questions on authentication, resource management, and degree of system integration that are affected. It is hard not to think that the future belongs with the heterogeneous systems, using some kind of "open systems software architecture" that defines remote procedure call protocol or remote file protocol. Many people can then connect their system. I would emphasize making the protocol as simple as possible, at the expense of power; consider the ease with which people have changed the C library vs. the difficulties of dealing with the kernel or the shell. Simplicity would also, of course, help with the serious and completely unsolved problem of debugging.

What do we need in the future? I doubt either the NASA & Livermore experience (multiple Crays) or the UCSD or Toronto configuration (hordes of students and almost no hardware) is typical of the future. Suppose the standard 5M workstation (1 Mips CPU, 1 Mpixel display, 1 Mbit network, 1 Mbyte main memory, and 1 mouse) becomes routinely available as an individual terminal. In that case distribution for load-sharing is not going to be interesting; remote files will be interesting. More important, however, is that putting several CPUs in the station will be fairly cheap. What should they do? Well, they could handle communications, so maybe efficient protocols will become less important. They could also handle caching, a favorite technique to improve performance. But most important, we need a way to use multiple processors, in local environments, to improve the performance of individual jobs.

One of my old managers at BTL used to show a viewgraph that went "1959 BESYS; 1964 Multics; 1969 UNIX." The trouble was he had nothing for 1974 or 1979. UNIX was designed as a uniprocessor operating system; C was designed as a single-thread procedural language; and I think we need some research that starts from other premises. Otherwise something else is going to be the operating system of the future.
European UNIX System User Group Meeting
Cambridge University, 19/21 September 1984
Peter Collinson
Secretary

Introduction

I hesitate to start this report with the magic words ‘this was another good conference’, but it was. There were three components: technical sessions in the Chemistry Lecture theatre; industry sessions held in the University Arms Hotel and organised by /usr/group/UK; and a large exhibition organised by Network Events Ltd., a company specialising in (yes, you guessed it) running exhibitions. I did manage to get to the exhibition, but didn’t make the industry sessions, I will endeavour to get someone to write a report of what happened there.

This report is again helped vastly by the abstracts which were supplied by speakers before the event, for which many thanks. We hope that the papers submitted for the conference will be printed soon. It is also hoped that the Proceedings for Paris will be printed before the conference, so speakers can talk about other things and confuse us all.

Day 1 - 19th September

Well, this wasn’t Day 1 for me, I had spent most of the day before in committee meetings of various sorts. This meant considerable quantities of local ale the previous night. Nothing daunted, I managed to make the really early breakfast and start the long trek down to the lecture theatre. It didn’t rain - I had my lucky umbrella.

9.45am Opening of the conference
Richard Stibbs, Cambridge University

Richard had mostly domestic matters to talk on, the most important being where to find the best beer in Cambridge. He violently denied the rumour that Greene King had sponsored the conference, and said that ‘Abbot Ale’ was not a trademark of that well known footnote.

After that David Tilbrook had a few minutes to introduce the people who were chairing sessions. He had brought a cutting from Computer Weekly of 13th September. This was in their gossip column and reads (all the spaces are theirs):

Peculiar

ANYONE wanting to attend one of the scintillating technical sessions at the forthcoming European UNIX systems user group meeting in Cambridge has to apply to (wait for it): (qlton,ukc)! ucl-cs! nigel qlton! ist! dt.

Is it any wonder that the rest of the world thinks that Unix people are a bit peculiar. Unix? Yukk! (sic) Geddit?!

The item was signed Chad. Recent attempts to send mail to their quoted net address: Yukk! (sic) Geddit?! has resulted in a profound silence.
9.50am UNIX Europe
Vanni Papli, UNIX Europe Ltd.

This was really an opportunity for Mr. Papli to announce AT&T’s hospitality suite. More seriously, he briefly described the state of the art in UNIX Europe. It is still quite a small organisation of only 12 people. They are responsible for all UNIX licensing in Europe and will also be running training sessions.

9.55am Communication Product Announcement
Joanne Miller, AT&T Technologies

Abstract
This talk will describe the current set of COMMKIT[†] Networking Software products for UNIX System V. A directional statement of the future of UNIX networking will be covered that relates to the network architecture, the operating system and the COMMKIT line.

Comment
The major thing of interest to me was the announcement of the availability of the new uucp 'Honey Danber' which Brian Redman talked about at the conference in Nijmegen. It seems that it will never be on a System V tape — it will always be a separate product.

The talk boiled down to a statement of intent which said: AT&T will use agreed international and de-facto standards, will document the protocols and make this documentation available so that other systems can converse with AT&T products. They will also supply technical support for customers.

10.28am Development of the MicroVAX 1
Robert Short, DECWEST Engineering

Abstract
Last year Digital announced the first of a new family of computers called MicroVAX. MicroVAX is a proper subset of the VAX architecture intended for VLSI implementation. The first member of this family, the MicroVAX 1, was developed at DEC's engineering site in Bellevue, Washington, making it the first major DEC hardware product designed and developed outside of New England. The MicroVAX 1 processor includes a custom MOS VLSI datapath chip, a cache and translation buffer. This talk describes the MicroVAX 1 hardware development, including:

- The advantages of subsetting the full VAX architecture to allow a small machine to be designed.
- The goals of the hardware design team.
- The microcode/hardware tradeoffs necessary to provide good performance while still meeting the cost/size/power goal.
- A fairly detailed overview of the hardware architecture of the machine.

Comment
The aim of the exercise was to make a VAX on a chip with the same cost as the M68000. The time scale for the project was 18 months, which is a short development period for a project of this sort. The time scale meant that the machine was based on the Q-bus which so that a totally new bus was not required. It was also decided to simplify matters by removing some of the 'fancy' but little used instructions from the VAX architecture. The machine can run non-privileged VAX native mode programs which are able to run under VAX/VMS on other machines in the VAX range.

The removal of PDP-11 compatibility mode, the decimal instruction set and some of the string instructions meant that there was considerably less micro-code than a full VAX. Instruction decode is

† COMMKIT is a trademark of AT&T Technologies.
easier because addressing is simplified by the removal of the PDP-11 instructions, the decode looks at a single byte at a time. All virtual addresses are 32 bits.

The following decisions were made in order to get a 32-bit processor on a 16-bit bus. The code block size is a longword and block mode on the Q-bus allows the processor to read 2 words at a time. The CPU microcode does not know about the 16-bit bus and does not wait for writes to complete down the bus. There is an instruction pre-fetch FIFO to speed things up a bit.

Well, DEC had one of these in the exhibition running ULTRIX. It apparently is faster than the VAX-11/730 but slower than the VAX-11/750.

Coffee

11.20am Large Systems experience - System 370
J. Carl Hsu, AT&T Bell Laboratories

Abstract
This talk will discuss the problems encountered and the experience gained during the development of the UNIX system for the IBM System 370.

Comment
Indian Hill is the largest computer centre in Bell Labs. Disc storage is 10\(^{12}\) bytes! The organisation is a technology leader in computing and networking applications, providing standard UNIX systems to other AT&T sites and also generating add-on packages to the standard system.

With the IBM machines, the objectives were to put UNIX on the IBM 370 in order to provide the full power of large systems and provide a working system for the computing centre environment. The implementation was a joint AT&T/IBM effort and uses an IBM supervisor for low level functions. The software is standard UNIX and is mostly in C. The system will support 180 users.

P and V semaphores were used instead of sleep/wakeup because the IBM architecture allows processes to be arbitrarily blocked in the kernel.

11.48am The portability dream
Neil Urquhart, Sphinx Ltd.

Abstract
The range of microcomputers offering UNIX has grown and with it the requirement to transport existing software between operating systems. The difficulties in porting software is seen able to be classified as four categories: UNIX development; machine implementation; programmer techniques; and software dependency.

The development of the UNIX operating system, from Version 6 through to System V, is overviewed, with special note being made of their idiosyncracies. Discussion of the contribution of the Berkeley implementation is included together with its influence on the new release of operating particular features which influence the transportability of application software or programs.

Examples of idiosyncratic code are presented to illustrate differences between machines, UNIX implementations and programming styles. C is claimed to be a portable language; it is discussed, with examples as to how much it encourages programmers to deviate from the 'straight and portable'. Some of the UNIX tools which are known to avoid portability difficulties are illustrated together with their limitations. Software dependency is discussed and the difficulties it can cause are illustrated.
12.07pm Multi-processing on UNIX
Steven J. Buroff, AT&T Bell Laboratories

Abstract
This talk will describe the development of a UNIX operating system that runs on multiprocessor configurations. The system currently runs on AT&T 3B20 computers, but the ensuing discussion applies equally well to other machine architectures that support a multiprocessor environment. Existing user level C programs can run without recompilation on uniprocessor and multiprocessor models of the 3B20 computer, unless the program contains system dependent code (e.g., ps(1)).

The talk will cover the critical region solution that was used (semaphores), dead-locks and resource starvations, scheduling, the protection scheme used for device drivers.

Comment
The abstract says it all. The current system is a dual 3B20 system and runs 1.6-1.9 times a single processor system. The speed improvement is typically 1.7 with a mixed workload.

Lunch

2.31pm Interactive three dimensional molecular graphics under UNIX
C. Huang, Computer Graphics Laboratory, University of California

Abstract
The Computer Graphics Laboratory at UCSF was established in 1976 for research on the structures and interactions of proteins, DNA, drugs and other molecules of importance in biomedicine.

MIDAS (Molecular Interactive Display and Simulation) is a large interactive molecular modeling graphics package developed at UCSF under UNIX, originally on a PDP-11/70 with an Evans and Sutherland Picture System 2 in black and white. It now runs on a VAX-11/750, and provides a flexible tool for the study of small and large molecules and their interactions, taking full advantage of available interactive three dimensional color display capabilities on both the Evans and Sutherland Picture System 2 and Multi Picture System and eventually on a PS300. Bond rotation, interactive monitoring of several distances and `docking' with real time representations of molecular surfaces is well supported.

Among its more innovative features is an unusually coherent hierarchical database for storage of macromolecules which minimizes both storage space requirements and access time. The ‘tool building’ philosophy encouraged by UNIX has resulted in a well organized and maintainable program that is well suited to reimplementation on UNIX -based graphical workstations such as the Silicon Graphics IRIS. The lecture will be illustrated with color slides and a movie. Supported by US National Institutes of Health research grant RR1081.

Comment
I was particularly impressed by the real-time nature of the pictures which could be generated.

3.08pm Does Darth Vader still program in Fortran? (or what do they really do at Lucasfilm)
Sam Leffler, Lucasfilm

Abstract
The original question posed by Bill Reeves at the USENIX meeting held in Boston, Massachusetts was ‘Does Darth Vader program in C?’; as one can see by the title the answer was clearly ‘No, he programs in Fortran.’ This presentation will attempt to update the audience on this noteworthy subject, as well as items of similar importance. High quality audio-visual material will be prominent in the presentation in an attempt to distract the audience from noticing that this talk is almost completely content-free.

For those that don’t believe the previous paragraph, the presentation will concentrate on describing what Lucasfilm does, how they use computer technology, and, most importantly, how UNIX is an integral part of everyday work, from the mundane to the exotic. Slides and 16mm film will be shown
of the most recent work from the computer graphics project.

Comment

Sam's answer to the question:

Does Darth Vader still program in Fortran?

was

No, the Ewoks do it for him.

Lucasfilm is split into several sections: Administration; Industrial Light and Magic, which is responsible for special effects; Skywalker Development Co., responsible for building a 'ranch' which will be used for filming; Sprocket Systems which does post production work on film; and the Computer Research and Development, where Sam works.

Sam talked about the computer based system which is used to fuse together several images in a process called 'blue-screen matting'. This is where separate films of different objects are taken on a blue background and the images are joined together eliminating the blue. Previous systems used a special machine where the original images are projected and a new negative made. Lucasfilm now have a system where the images are joined using a computer system.

Another area of interest at Lucasfilm is the use of computers to synthesise images. Sam showed a number of astounding slides, the best of which (or at least the one which sticks in my mind) was a picture called '1984'. This showed four pool balls on a table being struck by a cue ball. The pool balls had the obvious numbers. The four balls had all moved and the picture showed their motion blur. Sam pointed out that the high-lights in the balls (pin-points of light) contained the reflections of the room around the pool table. The picture was really great, and very hard to describe.

Sam talked about 'Andre and Wally Bee', a 2-minute animated film generated totally by computer. He was going to show the film, but the projector broke down, so we had an early tea. The film was shown when the projector bulb had been replaced, it has everything: three dimensionality, motion blur, forests generated automatically, to name just a few features.† Great stuff.

Tea followed by the film

At this point, I decided to visit the exhibition and skip the remaining talks for the afternoon. Everyone at the conference had to make the decision about what talks to miss so they could get to the exhibition. The program organisers had thought that two hours would be sufficient time for most people to go in the lunch break. However, they hadn't reckoned on the eons which it took to have lunch in St. Catherine's.

So, I left and made my way along the road to the exhibition hall. On my way past the RAC sign saying UNIX SYSTEMS 84, I began to feel that EUUG had come a long way since those 20-odd people met in Scotland those several years ago. Still...

The exhibition was large, at least by previous standards. Most exhibitors were in blue booths, obviously supplied by the organisers, Network Events. It felt very professional, and that was no bad thing. Certainly, the exhibitors who I talked to felt that it was OK, and since there were 2500 visitors, it must be adjudged a success.

One of the things which caught my eye was the SUN system with the wide screen. This was fun to play with and is the way I want to talk to computers.

At the low end of the market, Torch's Unicorn sitting behind a BBC computer is still the cheapest UNIX system you can buy. However, it really is VERY SLOW. The salesman told me that Torch is bringing out a system with a M68000 running at twice the speed of the current version. So, if you're thinking of buying a Unicorn, wait. Of course, the other problem is that the system is UNIX System III 'with the usual Berkeley enhancements'. The visible bell in vi was fun on the colour screen, giving bands of different colours.

† And you didn't need to worry about -v, either.
Possibly, the smallest machine, in physical terms, was the Spirit from UNIQIX Ltd. This looked an interesting product.

Anyway, I can't do real justice to all the exhibitors, so I'll stop. Let's hope that most of them come to Paris, so that perhaps there will be more time to look at them.

Meanwhile, back in the Chemistry lecture theatre:

4.10pm UNIX user interfaces for applications
Stephen Travis Pope, BST - Basic Software Technology Dept.

Abstract

All computer applications need some man-machine interface method. UNIX is especially well suited as a base for applications because programmers have access to system utilities and can substitute their own front-ends for or on top of the 'shell' program. Developing new user interfaces that are tailored to particular applications requires, however, that the designer repose several basic questions about the desired application, its users and computer I/O in general.

Computer configurations involving bitmap terminals and mice can also be used for very-high-level interfaces and programs can be built to take full advantage of these features. So-called 'window systems' as seen on the Xerox Alto, Lisp machines and the Apple Lisa computers can also be implemented in multi-user UNIX environments with excellent results.

Several current projects being undertaken at the BST dept. of PCS GmbH will serve to demonstrate special, window and/or menu based user interfaces for applications in program development, databank management systems, networking and computer music. Topics of this part of the discussion also include the hardware for the current PCS multi-processor implementation of the *sh window shell.

4.35pm Winnie: a new multiple window screen editor
Patrick Amar, United Software Artists Inc.

Abstract

We shall describe a new full screen editor called Winnie. Winnie belongs to the Emacs family. It has two major enhancements over Emacs: a more flexible windowing discipline and a multi-language extension facility. As opposed to Emacs, we have strived to build a small and efficient implementation. Winnie uses no more than 35 Kbytes of storage for code (compared to 121 Kbytes for Emacs on VAX) and loads the computer much less than Emacs.

Winnie can handle arbitrarily many windows of any size and placed anywhere on the screen. Windows can hide each other as sheets of paper on a table, or as screens in the Xerox Star station or the Apple Lisa machine. For the present time we handle only alphanumerics screens, but a graphic screen version is forthcoming.

5.05pm Questions and answers
Vanni Papl, UNIX Europe

As I had left, I asked Sean Leviseur from UKC to write some notes. So, this bit is from him.

Q. When will virtual memory support be available for System V?
A. Virtual memory will be available in the last quarter of 84 for System V.2.

Q. What will be in System V.3?
A. Chiefly file locking and support for virtual memory and paging.

Q. Will Edition 8 be generally released?
A. No.
Q. How will the presence of UNIX Europe affect pricing?
A. European prices will be the same as in the States.
Q. What price will the upgrade to System V.3 be?
A. This has not yet been decided.
Q. Will Edition 8 be released to educational institutions?
A. It will only be released under strict supervision to six American universities.
Q. When will the UNIX manuals be stabilised?
A. With the next release. They are currently being restructured, only one set is currently available. It would be useful if people could comment on the current System V manuals.
Q. Will System V be unbundled?
A. It is intended to unbundle more of UNIX but not until next year.
Q. Will the information flow from the States improve with the existence of UNIX Europe?
A. We will have more people from the States, so hopefully we should get information quicker.
Q. What about European availability of the Blit and its software?
A. The Blit is sold by Olivetti in Europe, although it is still available from AT&T’s old distributors in Europe. The software is available from UNIX Europe, as is the source code.

End of Day 1

And some of us went onto AT&T’s hospitality suite to obtain the odd alcoholic beverage; and then onto the Reception for all conference attendees in the University Combination room, where even more of the you-know-what was consumed.

Day 2 - 20th September

The lucky umbrella didn’t work this morning, the heavens opened and tried to wash Cambridge into the River Cam.

9.33am MMUs & the UNIX kernel
Robert Jung, Root Computer Ltd.

Abstract
UNIX system performance depends on many factors, one of which is the memory management unit (or MMU). Porting the UNIX kernel to Motorola 68000 and 68010 based systems has given Root a unique insight into the implementation and performance of MMU designs. This talk describes the kernel’s interaction with the MMU, observations and recommendations of MMU design and implementation, and a brief discussion of the memory-management schemes Root has come across.

Comment
Root has done many ports of the UNISOFT UNIX system. They have also used a number of MMU’s but the talk centred on comparing the Stanford MMU with the Motorola 68451 MMU. The Stanford MMU is also called the SUN MMU, and has nothing to do with the company of the same name.

The Standford MMU is better for memory allocation, switching processes and accessing the user area of the current process. But it is worse at accessing other processes, which is used for swapping. The Stanford MMU is cheaper and runs faster than the M68451.
10.01am **Paging in the UNIX system**  
Steven J. Buroff, AT&T Bell Laboratories

**Abstract**

Two research derivatives of the UNIX system have supported paging for several years: Reiser 32V, and BSD. Work is under way at AT&T Bell Labs to bring together the features of both of the systems (and others) to form a demand paged kernel for UNIX System V. This talk will discuss three areas of this work: requirements, architecture, and implementation.

**Comment**

The requirements are: there should be no user program changes for either binary or source; the system must not hurt users who don’t require paging, this means that if you want to use a paging system because it is faster, then you can do so; and the system should provide the capability of large address spaces if that is wanted.

The idea was to generate a general model of memory management in the UNIX kernel and to abstract all the code which deals with it into one generalised set of routines. Different memory allocation methods can then be used because a clean internal interface has been designed. The main primitive in the design is a *Region*, which is an area of memory. It can be shared or private and is manipulated by a set of well defined operations. These operations are: create, delete, attach, detach, grow, load and copy. The system allows copy on write by adroit use of the page descriptors.

The current implementation is for a paging kernel which appears to work. There didn’t seem to be a swapping implementation as yet.

10.28am **Productizing (zic) UNIX**  
Armando Stettner, Digital Equipment Corp.

**Abstract**

DEC is now offering a UNIX product. This talk will discuss some of the problems that were encountered when creating that product.

**Comment**

At the time of the decision to supply and support UNIX, it was decided to base the system on 4.1BSD. The system was fast and flexible and had support for many peripherals, so there would be much less work for DEC to do in preparing it to be a product. Also, at that time more VAX’s ran 4.1BSD than any other system. The intention was to switch to 4.2BSD when it came out.

The goals of the product were that it should be as least as reliable and predictable as 4.1/4.2BSD. Also, it should be compatible with those systems. DEC didn’t want to introduce yet another flavour of UNIX. DEC have not altered anything in the program execution environment and have only altered one user program (*tar* now follows symbolic links).

There were several questions:

Q. Can you install user written device drivers?
   A. Yes, the system is supplied in a configurable binary with important files being supplied in ‘un-compiled’ form.

Q. Does ULTRIX have the device drivers which control non-DEC devices.
   A. Yes, anything on the 4.2 distribution is also on ULTRIX. But, it will obviously be harder for DEC software support to provide advice and bug fixes on the ‘foreign’ peripherals.

Q. What does the management think about that?
   A. ULTRIX must fulfill the expectation of what UNIX does. So, the foreign device drivers must be supplied because they are part of UNIX.

---

\* At some point in the conference, I was told by a DEC person that 25% of VAXes in the UK are running UNIX.
Q. Can ULTRIX support the new DEC device clusters?
A. Not at present.
Q. Does DEC intend to move to System V?
A. No.

Coffee

11.17 am A secure high-speed transaction protocol
Sape J. Mullender, Centrum voor Wiskunde & Informatica, Amsterdam

Abstract
Most computer networks use a byte stream protocol for communication between processes, which suffer from two important drawbacks: the addressing mechanisms provided often are process-dependent or location-dependent, and communication is slow. While carrying out research into distributed operating systems at the Vrije Universiteit and the Centre for Mathematics and Computer Science, we have developed a transaction-oriented transport protocol, aimed for high-speed, with an addressing mechanism that is not only more general, but provides a protection mechanism as well. The basic mechanism for communicating between processes is the transaction: a client process sends a request to a server process, which carries out the request and returns a reply. Protection is provided by using ports, chosen from a sparse address space, for addressing services. These ports serve as a 'capability' for communicating with the service. Through its simplicity, the transaction protocol can achieve high transmission rates (more than 200 Kbytes/sec process-to-process, eventually).

The protection mechanism will be described, and the mechanisms for realising high transmission speeds.

11.48 am Connecting UNIX systems using a token ring
Robbert van Renesse, Vrije Universiteit, Amsterdam

Abstract
As part of the research on distributed operating systems being done at the Vrije Universiteit, we have implemented a set of network-oriented programs for use on several UNIX machines connected by a high-speed token ring. With these tools it is possible to transfer files between machines, log in to remote machines, and implement multimachine shell scripts. The transaction protocols discussed in another paper at this EUUG meeting are used to implement two basic services: a 'shell server' and a data transfer service. Other services are easily implemented as shell scripts that use these services. A file transfer program, for instance, executes the command to < file1 on one machine, and from > file2 on the other machine. More examples of these facilities and their implementation and performance are discussed in the paper.

Comment
Some real process-to-process throughput figures were mentioned: VAX/VAX - 25Kbytes/second and PDP-11/PDP-11 - 10Kbytes/second.
12.08am A project development environment for UNIX
Malcolm Crowe, STRG Paisley College of Technology

Abstract
A software environment for project development should include tools for all phases of the development process. Many such tools already exist, or are under development, in the UNIX system.

However, apart from archiving systems such as SCCS, little support exists in UNIX for quality and project management. A key activity of quality management is concerned with the identification and control of all items produced during software development.

Until recently configuration management has been an essentially manual activity, possibly performed by a member of the project team. In this paper, we describe facilities which allow project management to implement a configuration management plan on a per-project basis.

The basis of the system is an enhancement to the UNIX file system (not affecting the kernel), which does not alter the user interface for naive users. The resulting environment retains all the UNIX tools, while allowing for their use to be restricted in various ways on ‘controlled’ objects.

The system is available to other participants in the Alvey Programme.

Comment
This was an interesting idea which uses the C library to alter the nature of the file system. Files can now have: long file names (even on V7); a project defined set of attributes; multiple versions; and controlled access. The file system also supports the notion of projects and project hierarchies. The file attributes are defined by name and act rather like shell environment variables.

Lunch

Jim McKie who was chairing this session got some biographies, official and unofficial, which he used to introduce the next three speakers.

Mike Karels


Unofficial: In 1982, the Paris EUUG conference was blessed with the attendance of Bill Joy. His performance at that conference gave us the phrase ‘Paris mode’. Sam Leffler was brought over to the Bonn conference to apologise, and did such a good job that he also attended the Dublin conference. Due to his sterling performance in Eire, we required two UCB people, Eric Allman and Kirk McKusick to compensate at the next conference. This conference is please to have Mike Karels and we have yet to decide who will apologise for him.

Tom Killian

Official and unofficial: Tom Killian began his career as a high-energy experimental physicist but was unable to convince his colleagues of the value of Computer Science. Since 1983, he has been with the Computer Science Research Center at Bell Laboratories, where he has successfully dealt with painful childhood memories of MVS and SCOPE.

Greg Chessen

Jazz drummer: CC Riders 1967-9, Woody Herman Orchestra 1969-70; B.S Math, Union College, New York 1972; M.S. Computer Science, University of Illinois 1975; Ph.D. Computer Science, University of Illinois 1977; Member of the technical staff at Bell Labs, 1977-1983; Chief scientist at Silicon Graphics 1983 onwards. At Bell Labs, Greg was responsible for V7 multiplexed files; line disciplines, character
drivers and boot programs in V7; packet drivers; circuit simulation, PLA, board layout software in UCPS; design and simulation of Datakit protocols; and the design and implementation of the B-machine 10mips processor for Bell. At Silicon Graphics, Greg has done the XNS network software for 4.2BSD, System V and VMS.

Unofficial: Greg Chesson, known as Bambi to his friends, has made contributions to the UNIX world, none of which should be mentioned prior to his presentation to ensure a reasonable reception. He is well known for his ability to find tone-deaf band leaders who let him exercise his other talents which are normally executed with the tact and diplomacy for which he is well known. His hardware skills are second to few. It is said that there is yet to be a machine which Steve Johnson couldn’t overload or Greg couldn’t break. Greg enjoys sports that don’t involve standing; and the only skill about which he shows any modesty is his extraordinary imitation of a synchronised swimmer using Datakit.

2.36pm Life after 4.2: measuring and improving the performance of 4.2BSD
Mike Karels, CSRG, CSD, EECS, University of California, Berkeley

Abstract
The 4.2 Berkeley Software Distribution of UNIX for the VAX includes a number of new features and facilities that substantially increase its utility. However, it has several problems that can severely affect the overall performance of the system. These problems were identified with kernel profiling and system tracing during day-to-day use. Once potential problem areas had been identified benchmark programs were devised to highlight the bottlenecks. These benchmarks verified that the problems existed and provided a metric against which to validate proposed solutions. This paper examines the performance problems encountered and describes modifications that have been made to the system since the initial distribution. It also describes other work underway or planned at Berkeley.

Comment
Mike started with some general observations on 4.2. First of all, it seemed slower than 4.1, and the system throughput was down by about 20%. The many new servers added considerable system overhead. The new fast file system altered the workload characteristics, so that processes which were previously disk bound were now processor bound.

The system was measured in order to get some idea of what was happening. The results showed that the micro operations were about the same speed as 4.1, although pipe and exec were a bit slower. The measurements also showed that name translation was about 40% of the system call overhead. Symbolic links add a measurable overhead.

There is some mileage in improving some user level programs. For instance, programs accessing the password file can be made to go faster. The standard I/O library has been altered to use optimal buffering.

In the kernel, the name lookup routine has been made 60% faster by use of caching. This results in an 8% improvement in the total system time. The dz and dh drivers use the silo for slow transfers, which loads the system. The code has been altered to use a single interrupt for the intermittent transfers which constitute most terminal input. The clock interrupt routine has been made to work faster. The arguments to the exec system call used to be read into the kernel a character at a time, the new system uses the much faster block copy code to read in strings. The context switch code has been made to run faster. Pipe performance has been improved by supplying more buffering. There have been several other minor alterations.

The system will be available to the public before December 1995[?].

New things which are being worked on a UCB include: a network file system which connects systems using a single tree using a remote mount system call; there is work being out into protocol layers; and a reliable remote procedure call mechanism.

At the end of his talk, Mike made a presentation of a game of ‘Battlecars’ to Armando Stettner. Apparently, Armando is famed for his accident prone driving and has received many car (or perhaps 1
should say automobile) related presents at US USENIX conferences.

3.00pm *Processes as files*
Tom Killian, AT&T Bell Laboratories

**Abstract**
We describe a new file system, `/proc`, each member of which, `/proc/nnnnn`, corresponds to the address space of the running process whose pid is nnnnn. Access to these files is restricted, via the normal file protection mechanism, to the process owner. `lseek(2)`, `read(2)`, and `write(2)`, allow inspection and modification of the process’ image. Other services are available via `ioctl(2)`, including stop/go on demand, selective interception of signals, and the ability to obtain an open file descriptor for the process’ text file. The technical problems related to the implementation of `/proc` on a VAX under the 8th Edition of the UNIX operating system have mostly to do with the paging system. Security issues are also considered.

The window-based interactive debugger `pi` developed by T. A. Cargill, is the first major user of `/proc`. It can control multiple processes dynamically and asynchronously. Thanks to the network file system, `/n`, these processes may be running on several different machines. We also describe an efficient, almost portable `ps(1)`.

**Comment**
This is such a reasonable addition to the file system name space, it took a genius to think of it. `Pi` is based on the Blit terminal.

3.31pm *Multicast ring protocols for real time games and other useful pursuits*
Greg Chesson, Silicon Graphics Inc

**Abstract**
Many great advances in computer science have been motivated by things that some (though not all) would deem as unimportant. This talk is about the solution to the less than pressing problem of being able to ‘fly’ multiple flight simulators in formation. Only the future will tell if this solution is a great advance, but it is brought to you by the implementor of multiplexed files and this conference’s shoo-in for the hairy knees contest.

A demonstration (on video tape) will be shown.

**Comment**
This is fairly verbatim from Greg’s initial slides:

**WHAT**
do we want to do?
To use an Ethernet as a token ring.

**WHY**
do we want to do it?
To synchronise real time processes.

**MOTIVATION**
to transform a real-time flight simulator on UNIX into a dogfight program for several machines running several programs controlled by several ‘pilots’.

Features of the flight simulator include the ability to choose one of several aircraft - Cessna 180, 747, F15, F18 and F16. Weapons on the aircraft include Sidewinder, rocket and cannon. Pilots of the Cessna spent most of their time circling the airfield and picking off other people as they take off. The output is in 3 colours with a 1K resolution screen, input is by mouse - no joystick yet.

The problem is how to get one program running on one machine update all the other programs on the position of their aircraft. The mechanism used is to broadcast datagrams rather than having a ‘star’ network of virtual circuits.

This talk was certainly one of the highlights of the conference. The video tape spend most of the conference winging its way in a non-simulated aircraft across the Atlantic. It reached Cambridge by
Friday evening, and was worth waiting for.

\[ \text{\textasciitilde} \text{Tea} \text{\textasciitilde} \]

4.31pm UNIX IPC: where it's been, why it left, where it might be going
Mike O'Dell, Group L Corporation

Abstract
Few topics in the UNIX community have provoked as much discussion and as many implementations as the issue of providing 'good' interprocess communication (IPC) for UNIX. The problem, of course, is not with pipes, for everyone agrees they are wonderful. The problem of interest is a scheme whereby unrelated processes can rendezvous and communicate. Indeed, it is a safe guess that any IPC scheme which has appeared in print has been implemented at least once in some UNIX system somewhere.

If there have been many implementations of IPC for UNIX, why is it that none of them ever seem to catch hold and be adopted widely as a general mechanism? Why indeed!

This talk will attempt to provide some historical background by reviewing some of the more influential IPC implementations, and discuss the author's view of the question posed above. Finally, the author will propose a model for unifying two important stream IPC facilities, and make some speculations as to how this unification might point the way toward a 'natural embedding' of IPC in UNIX.

Comment
This was one of the best talks of the conference. I feel that I cannot do any justice to it from my notes - you will all have to wait until the proceedings are published. Suffice it to say that the talk gave a really comprehensive review of the many different schemes for IPC with an idea of the problems and advantages of each.

\[ \text{\textasciitilde} \text{End of Day 2} \text{\textasciitilde} \]

Well, via AT&T's hospitality room to the conference dinner, which was a pretty splendid affair. After the dinner, The Instruction Set did a side splitting selection of sketches. The EUUG committee were asked to sit in the front, we were bearing up for custard pies or something of that ilk, but nothing really nasty happened - we just laughed a lot. I also met Mr, Mrs and Miss Tis who were very nice.

Day 3 - 21th September

9.33am Implementation of OSI protocols under UNIX in the EIES network
J. Loveluck, Bull

Abstract
The Esprit Information Exchange System (EIES) is an infrastructure to support collaborative R and D projects in information technology within the European Study Program in Information Technology (ESPRIT) launched in 1984 by the European Economic Commission. The work is being carried out by a consortium of 6 industrial partners.

The EIES will provide the electronic mail, teleconferencing, document handling and transfer, file transfer, remote login services which are necessary for cooperative R and D work.

The project aims at a maximum connectivity of potential users through the use of Open Systems Interconnection ISO services and protocols, and starts with an implementation under UNIX.

After a short description of the objectives of the project, the paper describes the detailed architectural choices made for the interconnection of local area networks and wide area networks; the addressing scheme is discussed, and some considerations given to the management aspects of the network. Finally the main choices made for the implementation under UNIX are described.
10.12am The Instructional workbench: a CAI system and more
Thomas B. Reddington, AT&T Bell Laboratories

Abstract
Any computer-assisted instruction (CAI) system must allow an author to easily create courses for producing on-line dialogue with a student. Instructional Workbench (IWB), the CAI system implemented for the UNIX operating system, is particularly efficient for building a friendly human-machine interface and, therefore is a strong alternative to other CAI systems and programming languages.

Two important features of IWB that will be discussed are the design of the authoring language and the authoring system that enables novices to 'mass-produce' on-line dialogue programs (courses) without requiring a detailed knowledge of the authoring language.

A production system served as the model for the authoring language of IWB, called TOPIC language. This design has proved to be particularly useful for defining the logic inherent in complex human-machine interactions and for easing the work required to modify that logic. Features which are 'built-in' to the TOPIC language are:

- I/O from files
- keystroke verification of user input using regular expressions
- terminal independent screen management capability.

The design of the TOPIC language has been so successful that parts of IWB that interact directly with a user are written in the TOPIC language.

The design of the IWB authoring system was driven by the needs of the intended authors: subject matter experts rather than programmers. The authoring system is template-based and allows authors to produce on-line dialogue programs quickly by filling in the 'holes' of the templates. The templates, which are data driven programs coded in the TOPIC language, allow an author to build dialogues by specifying the data particular to an interaction. In CAI common types of templates are true/false questions, multiple choice questions, and tests. In addition the authoring system can be extended by the addition of templates written by a programmer with some familiarity with the IWB TOPIC language. This robustness has enabled the authoring system to be useful in areas outside of CAI such as on-line help systems.

The presentation will also discuss other features, such as computer-managed instruction, that make IWB more than just a CAI system.

Comment
This is brand new and is not yet a product. I think that it looks very good.

10.34am What is happening at Pyramid
Robert Reglan-Kelly, Pyramid

I am afraid I missed this talk.

Coffee

11.22am UNIX in Australia, 1984
John Lions, University of New South Wales

John started by talking about the history of UNIX in Australia. He then spoke of some of the work which Tim Long has done in making the file system faster. This involves keeping the inode list of free disc blocks in ascending order; varying the look-ahead from between 1 and 32 blocks depending on experience; and finally coalescing requests for I/O into one large request. John then spoke about the scheduling system which Sydney University uses to get 100 students working on a VAX. The mechanism is based on the user having a share of the resources in the machine; the system is called MUSH and is where UCB got their disc quota system from.
The Australian alternative to the *uucp* network is called ACSNET. This was originally called the Sydney UNIX Network, or SUN; but unfortunately this name has been stolen by another company in the UNIX world. The name is still retained for the software. ACSNET is a message passing service involving multistage transfers but with no explicit routing. We are promised more on the system when Piers Lauder comes to Paris.

John noted that there was a general movement from DEC hardware in Australia; he mentioned that Melbourne was very happy with the Pyramid.

John finished which a few slides: one was of a wallaby reading the *Kernighan/Pike Book* - and very fetching it was too.

11.55am **UKUUCP and other EUUG software**  
**Lee McLoughlin, Westfield College, University of London**

I have the benefit of Lee's overhead projector slides for this - Ta.

UKUUCP is a single *uucp* combining the best features of the dozen or so *uucp* versions found in the UK. It includes work done by Mike Bayliss, Tony Luck, Jeff Smith, and Lee himself. UKUUCP includes hooks for dial-in/call-out on the same line, on V7 and 4.2BSD and has loads of bug fixes. Its performance is enhanced by 40-fold by improving the scanning of queues.

The code should port easily to other systems. It is currently running on VAXes, PDP 11s, LSI 11/23, various 68000s and the High Level Hardware Orion. The code is running in the following UNIX systems: V7, 4.1/4.2BSD, System III and System V.2. The code has failed on GEC 63's and Pergs.

The main claim to fame is that it can transfer over very simple connections, such as PADs, GECs, PRIMEs, and other systems which provide only tty facilities for networks.

UKUUCP needs York Box support (this is underway); X.25 support as in System III/V; and many other things.

The code is available on the EUUG tape. You require a V7 license (or better).

12.11 **EUUG business meeting**  
**Emrys Jones, EUUG**

The main business was to report that there had been one written submission about the proposed constitution with syntactic alterations. The constitution will be sent to all members for a postal ballot in the near future.

Actually, there is also one matter of report which Emrys did not mention but which I thought I would slip in here. At the Committee meeting in Nijmegen, it was decided to award an honorary membership to Alan Mason in recognition of his work in laying the foundations of the EUUG.

2.30 **Speech input and UNIX**  
**R. M. Johnstone, University of Glasgow**

*Abstract*

This work investigates the performance of speech input in certain application areas.

The preliminary work involved selecting a task which seems particularly well-suited to use of speech input, and building a speech system around the task (using a UNIX host), following principles derived from previous work in the area). Low-level recogniser functions are controlled by a set of simple C programs. Vocabulary are stored as UNIX text files, for re-use in the next speech session. The facilities of *emacslisp* were of considerable use in implementing and maintaining dialogues for particular tasks.

Initial experiments involved the experimental manipulation of task variables (e.g. operator experience, system training procedure, vocabulary composition), and certain recogniser parameters. This type
of work aims to produce guidelines for applying speech recognition, which will enable a system to be optimally tuned.

Our current project looks at how well speech recognition can replace keyboard input. Once again, our answer will depend on a large number of variables, including user cooperativeness and experience, and characteristics of the particular task. Candidate tasks include Pascal programming and nroff command insertion.

2.57pm Measuring disc I/O on the VAX
Nick Nel, University of Glasgow

Abstract
This paper describes a project under way at Glasgow University to gather statistics about disk performance on a VAX running Berkeley UNIX 4.1. These results will be used to construct a stochastic model for the behaviour of the disk subsystem. We hope that by modifying the parameters on the model and studying the results we can discover new ways of improving the disk and file system performance.

Comment
Nick was rushed when he talked about this at Nijmegen and was given more time to present the results again. The main idea was to measure disc performance in order to find useful measures of spread and central tendency of disc traffic, i.e. the mean arrival of requests for disc transfers and the pattern of requests. Armed with the mathematical model, it should be possible to predict future trends and generate reconfiguration forecasts.

The measurements taken were at the start of the strategy routine, which is the point where UNIX says 'OK here's something to do on the disc' to the relevant disc driver.

The measurements showed a generally exponential distribution but there are two interesting peaks: one at a few micro seconds and a one at 20ms. Why these peaks exist is still being investigated.

3.20pm sndawk - a signal processing language
Dan Timis, Institut de Recherche et Coordination Acoustique/Musique, Paris

Abstract
Signal processing and sound synthesis often use pipelines of programs performing specific treatments as filtering, changing the sampling rate or the gain etc. on binary floating point samples. Users who want to make their own algorithm will spend sometimes, for a very simple thing, time and energy to write, to compile and to test a program in C or Fortran.

Sndawk (sound awk) provides an interpretative signal processing programming language simple to learn and to use. Inspired from the well known pattern scanning and processing language awk, it respects much of its syntax as it respects much of the syntax of C.

This talk will discuss the structure and use of sndawk

4.30pm Pontifications, Accusations, Prognostications & Mystifications
Chaired by David Tilbrook, Imperial Software Technology

Abstract
This session will be an open forum for discussion of UNIX and its future. Each panelist will make a 3 minute presentation on their views and prejudices, after which the floor will be open for questions,
comments and discussion.

Comment
This session is impossible to take notes on and luckily Richard Stibbs had organised a tape recorder to record all the important noises with. I had not thought of that and will certainly do it again, — thanks for the idea, Richard.

The session started with a couple of small presentations. The first was introduced by David Tilbrook.

"I don’t want to embarrass people by asking: who has an illegal copy of John Lions’ book?† But when I put out the advanced notice saying The much-xeroxed John Lions, I got an enthusiastic response. It was a remarkable book, we owe him a lot, and we would like to give him a small token of our appreciation in lieu of royalties." The present was a pint beer mug bearing the arms of John’s college, engraved with To John Lions, in appreciation from the EUUG, Cambridge, September 1984.

David was then presented with a large pair of big red inflatable lips by Jim McKie, who said: ‘‘There is someone here who has done a lot of work for this conference and it’s often been said that he has the biggest mouth in the UNIX world - so here’s the biggest lips to go with it.’’

The proceedings then really started with each member of the panel making a three minute presentation.

Armando Stettner - DEC

‘‘From its beginning, UNIX could never be considered a ‘standard’ operating system, or rather it has always been considered to be evolving. Once UNIX allocated its first inode it started to evolve. Its evolution was in the hands of a very few people; beginning with a small group in Bell Labs, Murray Hill and moving onto small groups in Universities around the world. There were other groups inside Bell Labs who were interested in turning UNIX into a tool for research and development of applications. Berkeley got into it, and did the right kind of things with 3BSD and 4BSD. Or at least in the early days between 3BSD and 4.1, they did the kind of things which were needed.

Now we’re into new era, UNIX in a larger sense will no longer be driven, or certainly no longer controlled, by purely technical and research people. Thankfully, I believe, 4BSD will continue to evolve along the path that makes sense.

Perhaps unfortunately, UNIX evolution is now in the hands of the corporations, IBM, Perkin-Elmer, AT&T, (DEC? from the audience), DEC, yes. These people will be driven by their perceptions of their customer’s perceptions of what they think they need. I can only hope that these new features and capabilities, and the functionality that the companies implement will fit into the framework and architecture of UNIX. I feel that this is part of the job of every UNIX guru and wizard who work for these companies that sells or distributes UNIX. Hopefully, the UNIX architecture will also evolve to facilitate its use on new technologies and new ways of building systems.

With the new kids on the block, the corporations, I don’t know where UNIX family is going. I can only hope that it will evolve and those of us who are involved in its evolution will keep an open mind for new things and new values.’’

† John Lions produced a book, or rather two books, describing the workings of UNIX V6. They were aimed at teaching undergraduates about the internals of the operating system but ended up training nearly all the UNIX hackers who existed at that time. AT&T sat on the book, only allowing one copy per site. To prevent any repetition of the incident, licencees of V7 were prevented from teaching the operation of the kernel.
John Lions, University of Sidney

John started his talk with a visual demonstration of C. P. Snow’s law: every culture is composed of two subcultures. This doesn’t translate easily onto the printed page. He wanted to talk about the small group of people

“... who can carry forward the true UNIX tradition. Which is not only to take two steps forward all the time. But occasionally to take one step backwards; and not only to add a few new features to the system; but also to remove some redundant, over-grown, over-ripe features from time to time. I think that if we all took a vow to declare 1985 the Year of the Pruning Saw and spend a lot of time cutting back, removing things which aren’t really needed; then UNIX may thrive. Without this, it will grow and nobody can do what Ken Thompson and Dennis Richie did for so long, namely, cut things back. If the unimpeded growth is allowed to continue, then the UNIX tree will die in the foreseeable future.”

Tom Killian, AT&T

“It says on my badge that I am from AT&T but I would like to issue a disclaimer: The opinions expressed are those of the speaker and do not necessarily reflect those of AT&T, its lawyers, cupholders, inquisitors, or anybody else for that matter.

Some of the questions brought up recently at this conference have been to do with the question ‘What is UNIX?’, and I propose a number of possible answers to this.

- A trademark of AT&T Bell Laboratories.
- It ought to be a trademark of Brian Kernighan, since he came up with the name.
- Is it a kernel?
- Is it a set of standard utilities?
- People watching this conference from the outside are probably convinced that it is a cult subculture.
- It’s almost certainly a state of mind.
- In some sense, it’s also what runs on Dennis Ritchie’s machine.

I think the UNIX philosophy is summed up in something which was written a long time ago by William of Occam, I am permitted to quote it in Latin, since this is Cambridge (he then quoted it in English?): ‘Objects should not be multiplied unnecessarily’. This is something which has guided UNIX from the beginning. There were a very large number of decisions which were made early on by Ken Thompson and other people in his office writing things on the blackboard. By and large, the decisions were right and it’s very dangerous if you go in and mess with them. Such things as the file system and the simple kind of scheduling were very crucial to making UNIX as successful as it was on the machines which were available at that time. Obviously, some these things are going to have to evolve for UNIX to remain viable. But I think that the philosophy that you have a minimal spanning set of necessary operations is very important. If we lose sight of this UNIX will be cooked.”

Mike Karels, UCB

“First of all, I will try to describe what I see of where the system is going and how we are trying to get there. Then I shall pick up on what the previous speakers have said.

The model of the computing environment that we were getting looking at in Berkeley 4.1 and which formed the major driving factor in designing 4.2, was that the single machine timesharing system was probably not going to be the way of the future. We are looking at workstations connected by a network, most probably ARPANET, and hopefully to the rest of the world on long-haul nets. So there are a lot of communications facilities and networking in 4.2. To parallel that, there are interprocess

---

\footnote{He put it on the overhead projector in Latin and whipped it away before I had a chance to copy it.}
communication facilities. [Tape garbled] The new directions in 4.2 come from that. Although there are communications facilities, there aren’t any really distributed facilities in existence. I don’t think UNIX ever will be a distributed operating system, this is tempting nature. On the other hand, UNIX can provide facilities of a distributed nature. This is something which is just getting going.

Another strong feeling that I have about UNIX is that 4.2 has gotten to the point of being a very large complicated beast, not only in terms of user facilities but also in terms of the way things are done inside the kernel. One of my long goals has been to redesign the kernel unifying things. Most of this would be invisible from the outside, but its one of the things I’d love to spend a few years on.

Teus Hagen, CWI

'Some time ago, we had languages, and we started to say that we must standardise those languages so that we could port programs between systems. So, Fortran was a big effort and after a while the Americans discovered that structured languages were a good idea, and so we had C. C has become a little standardised now.

UNIX helped a lot in standardising operations on the machine. I don’t think that a lot of people involved in ‘standardising operations’ can make something which is portable.

4.2 was a lot of work, mainly in the area of networking and I think that’s one of the first problems. The networking implementation is very young, it’s one of the first so it has to be changed. The resulting system will not be UNIX any more. We have to wait until somebody pops up doing real networking. So, I think that in a little bit, UNIX can withdraw; and in the end we will have a real networked system. But I don’t know what that will be.’

Steve Bourne, DEC

‘At Silicon Graphics, I was considering how to make our business decisions safe against the vagaries of all the different technologies and groups. There are lots of different versions of UNIX out there, and if you were a small company like we were, which version do you chose? Well, of course, what you have to look at, is who the big guys are; or at least, who is likely to be leading the charge. What you have to look at is AT&T, IBM, DEC and (Berkeley). I’ve put Berkeley in parentheses because they are not a large corporation, or at least not yet, but they are clearly a major force in making technological advances in UNIX.

I am not sure what you should conclude from it. I’ve spent some time at this conference nattering about how AT&T and IBM are facing off, and how AT&T are yet to have a well oiled marketing machine. Well, anyway so that was one question which I was interested in.

The answer, of course, isn’t very interesting. We must hedge our bets. What I mean by this is that we try to chose the subset of UNIX which we use so that programs would port. There isn’t much help for porting in terms of tools to look at libraries, and to look at porting difficulties; telling you whether your program is going to port from one place to another.

The question is should we accept divergence? UNIX never legislated anything, the group which made it never legislated anything. There are two more questions: can we as a group continue to make the right decisions and secondly, if we can, can the large group continue to make progress?’

General discussion

David then opened the discussion up to the floor. Sorry folks, you aren’t going to get a verbatim transcription of that for two reasons: first, I don’t have the time. Secondly, the tape isn’t exactly hi-fi quality and some of the floor contributions are totally lost in crackles and hiss. Much like conversation on the UK telephone system these days.

Mike O’Dell kicked off the discussion with a longish statement (most of which is lost). His main thesis was that UNIX makes a number of powerful statements about what one might reasonably expect
in an environment where programs are developed. It managed to do this by not writing a paper about it but by implementing it. He then moved on to say that people seem to think that putting a C compiler onto a system imparts some of the magic of UNIX. It doesn’t, a bad system with a C compiler is just a bad system on which you might be able to program in C.

The discussion then moved onto the marketing and standards in UNIX. There was general assent to the idea that restraining developments by the imposition of ‘standards’ was a bad thing. UNIX will always alter to fit the environment in which the system is to be run, whether this is done by companies or universities. The main hope is that whatever is done to the system is done well. One voice from the audience did plead for a single standard, even if it is a bad one, he saw that the problem was that there were too many dialects. A number of voices were raised against this view, the argument seems to be between those who think UNIX is an evolutionary process and believe that ‘official’ standards will inhibit this; and those who want and need a stable base for the development of applications software.

The question was raised as to whether you should run System V or 4.2 on a VAX? Most people in the audience said 4.2 - perhaps that says something about the audience. David then asked whether we should all be trying to run Edition 8 rather than 4.2. John Lions said that if Rob Pike was to be believed: the file system throughput under Edition 8 is considerably greater than under 4.2; the kernel is more compact; and line disciplines are a reasonable alternative to sockets. Cries of ‘not true’ from Mike Karels.

John Lions then moved the conversation onto the availability of bit-mapped displays. There was a general feeling that the cost would reduce to a level where they were affordable in the same way as a standard VDU is today.

Discussion moved onto how network file systems should be implemented. Tom Killian said that the strength of method where remote files were joined to the local file system as part of the tree was that all the current binaries just work, the namei routine in the kernel just works harder. Mike Karels endorsed this.

There was then a lot of discussion, with not many new points. David wrapped up the conference and we finally got to see the video tape of Greg Chesson’s flight simulator system.

Endpiece

We did, as usual, have space for writing silly comments on a blackboard. The idea was for people to write down the error messages which they found of least use excluding the ‘?’ and ‘TMP’ messages from ed. We got the following list, I leave it as an exercise for the reader to check whether they are true or not.

  local symbol botch   - V7 ld
  ERK! - (V6 passwd & V7), we had a later spelling correction to URK!
  oops   - termlib.
  Bailing out at line ... - awk, especially when the ... is replaced by 1.
  eh?   - chess
  Values of B will give rise to Dom - V6 mv. This is my favourite.
  Invalid keyword "else" - C compiler.
  Very funny - V6 pr.
  arge=2 - V6 comm.
  Clock may be set wrong - scs get
  mo40   - C compiler.
  Intruder alert - 4BSD whoami
Jackpot — V6 diff.
Room must be Cory or Evans — 4BSD chfn.
Termination code 132 — 4BSD f77.
ERROR
Illegal character 104 (octal) — pce.
Lints little mind is fried — V7 at least. I checked this on 4.2, and the correct coding for that
system is ‘Lints little mind is blown’.
No Toy clock — V6 date.
Modify failed — csh
Syntax error near line 1 — awk for a 1-line program.
Mere mortals mustn’t use that mantra — sendmail
What? — quiz
Well, that’s it. This file is now about 70000 bytes, which is too long. Thanks to all who made the
conference work.

USENIX Conference Proceedings Available

Proceedings for the following USENIX conferences are available from the organizations listed.
California residents please add applicable sales tax. Payments must be in US dollars payable on a US
bank.

Salt Lake City — Summer 1984, and Toronto — Summer 1983

Copies of the proceedings of the Salt Lake City Conference are available for $25 per copy, and of
the Toronto Conference for $30 per copy. Add $15 per copy for overseas postage. Send your check or
purchase order to:

USENIX Association
P.O. Box 7
El Cerrito, CA 94530

Payment must be received before proceedings will be shipped.

Washington DC UniForum Conference — Winter 1984

Copies of the proceedings of the UniForum Conference are available for $30 per copy, plus $20
per copy for overseas postage. They may be ordered from:

/usr/group
4655 Old Ironsides Drive, #200
Santa Clara, CA 95054

San Diego UNICOM Conference — Winter 1983

Copies of the proceedings of the San Diego UNICOM Conference are available for $25 per copy,
plus $15 per copy for overseas postage. Send your check or money order to:

Software Tools Users Group
1259 El Camino Real, #242
Menlo Park, CA 94025
The First Bimonthly UNIX Crossword Puzzle

Peter Langston

across

e0 On a PDP-11 test the N & V condition codes; if they are either both clear or both set then replace the PC with PC + (2 x offset).
d1 What a UNIX process does when it wants a clone.
c2 Prunus Spinosa (What does this have to do with UNIX? Well, some say it sounds like a description of 4.2 BSD.)
h2 How does it hyphenate words?
b3 Many people's unflattering idea of what "User Friendly" means.
a4 UNIX's travel command.
f4 Filthy, foul, wretched, vile, base, morally degraded, grasping (e.g. the details of TSO's workings).
a5 "— touch someone" (3 wds)
a6 The more communal counterpart to UNIX RT.
d6 Netnews caters to many appetites, but net—radio has no recipes.
j6 How to convert and copy ε file.
b7 What UNIX is.
c8 Control-M.
h8 See b7 across.
d9 UNIX Version in Jan. 1990 (if things keep going at the current rate).
e10 A place where your ttys are kept.

down

a4 High-tech device made economical by mass production for entertainment.
b3 File that defines legal BLICN destinations (careful!).
c2 Switch source file.
c7 Important shell script that no user ever runs.
d1 Largely a relic of the batch operation days.
e0 termcap capability type.
e5 The bigger Morse symbol (backwards).
e9 Program whose "1" command mishandled DEL in version 3.0.
f0 "Reach out and — " bumper sticker (2 wds).
g0 Your option to set erase and kill characters to # and @ in BSD.
g3 Really obscure UNIX acronym (sorry).
g9 Entertainment from a4 down.
h1 Advantage to having a command in /bin (2 wds).
i2 A popular culture role model for UNIX wizards.
i7 The way to cure disk space problems.
j3 From /usr/gooftball print all files under /usr/gooftball/k with: "— -print" (2 wds)
k4 Why AT&T loves uucp & netnews.
Local User Groups

The USENIX Association will support local user groups in the United States and Canada in the following ways:

- Assisting the formation of a local user group by doing an initial mailing for the group. This mailing may consist of a list supplied by the group, or may be derived from the USENIX membership list for the geographical area involved. At least one member of the organizing group must be a current member of the USENIX Association. Membership in the group must be open to the public.
- ;login: will publish information on local user groups. Information on local groups giving the name, address (phone number and/or net address), time and location of meetings, special events, etc. is welcome.

Please contact the USENIX office if you need assistance in either of the above matters. Our current list of local groups follows.

The Front Range group meets about every two months at different UNIX sites for informal discussions.

Front Range Users Group
N.B.I., Inc.
P.O. Box 9001
Boulder, CO 80301
Steve Gaede
(303) 444-5710
hao!nbires'gaede

Dallas / Fort Worth UNIX User's Group
Advanced Computer Seminars
2915 L.B.J. Freeway, Suite 161
Dallas, TX 75234
Irv Wardlow (214) 484-UNIX

There is an informal group that meets in the Washington, D.C., area every two months or so. The current contact for that group is:

Neil Groundwater
Analytic Disciplines, Inc.
Suite 300
8320 Old Courthouse Road
Vienna, VA 22180
(703) 893-6140
npg@lbl-csam

Unigroup is a non-profit organization in the New York City area for users and vendors of products and services for UNIX systems.

Unigroup of New York
G.P.O. Box 1931
New York, NY 10116

The UNIX Users of Minnesota meets on the first Wednesday of each month. For information on times and locations contact:

UNIX Users of Minnesota
Carolyn Downey (612) 934-1199

In the Atlanta area there is a group for people with interest in UNIX or UNIX-like systems:

Atlanta UNIX Users Group
P.O. Box 12241
Atlanta, GA 30355-2241
Marc Merlin (404) 255-2848
Mark Landry (404) 874 6037

In the Seattle area there is a group with over 150 members, a monthly newsletter and meetings the fourth Tuesday of each month.

Seattle / UNIX Group
P.O. 58852
Seattle, WA 98188
Irene Pasternack (206) FOR-UNIX
uw-beaver!tikal!ssc!slug
USENIX Searching For An Executive Director

As the USENIX Association continues to grow, it is becoming apparent that relying upon volunteer labor to manage the Association is no longer feasible. The Board of Directors has decided to hire an Executive Director to oversee the functioning of the Association.

The Executive Director will report directly to the Board and will be responsible for the smooth operation of the USENIX Association; major duties will include:

1. Supervision of member and office services and management of office staff
2. Responsibility for operation of the Association's computer
3. Supervision of conference planning
4. Supervision of newsletter and distribution tape production
5. Fiscal control, monitoring, and control of expenditures
6. Handling of minor policy issues
7. Handling public relations.

This role is expected to develop into a strong leadership position. Candidates for this position should have previous management and supervisory experience and should be familiar with the basic usage of computer systems, preferably UNIX. The candidate must have sufficient maturity to handle minor policy matters and work without supervision or direct guidance, and must be self-motivated, well organized, and thorough with attention to detail. The ability to interact and deal politely, tactfully, and effectively with a broad assortment of people is also required.

The position will be based at the USENIX office in El Cerrito, California. Anyone interested should send a resume to the USENIX office:

{ucbvax, decvax}!usenix!execdir
USENIX Association
P.O. Box 7
El Cerrito, CA 94530
Attn: Office Committee

Or, for more complete information, contact:

Deborah Scherrer {ucbvax, decvax}!usenix!scherrer
Tom Ferrin {ucbvax, decvax}!usenix!tef
Lou Katz {ucbvax, decvax}!usenix!lou
4.2BSD Buglist To Be Available As 84.2 USENIX Distribution Tape

mt Xinu, Inc. has arranged to contribute the 4.2BSD buglist to USENIX for release on the 84.2 distribution tape. mt Xinu has collected material submitted to 4bsd-bugs@BERKELEY (not from reports submitted only to net.bugs.4bsd, for example) and organized and edited it as a service to the UNIX community on a non-profit, non-commercial basis.

The buglist being distributed is essentially "raw". No judgment has been passed as to whether the submitted bug is real or not or whether it has been fixed. Only minimal editing has been done to produce a manageable list. Reports which are complaints (rather than bug reports) have been eliminated, obscenities and content-free flames have been eliminated, and duplicates have been combined. The resulting collection contains over 500 bugs.

Two versions of the buglist will be distributed:

All-but-Source:

All material except AT&T or 4BSD licensed source material. Nearly a mega-byte, this tape includes descriptions of bugs and the submittor's name; however, many of the fixes have been eliminated because they contained source-licensed material. This tape will be distributed to Institutional or Supporting members who have 4.2 systems but do not have 4.2BSD and AT&T source licenses.

All-with-Source (FOR SOURCE LICENSEES ONLY):

4.2 source licensees who also have a suitable AT&T source license will obtain a tape containing all the material, including proposed source fixes where such were submitted. This will be distributed to any Institutional or Supporting Member who possess both a 4.2BSD source license and an appropriate and verified AT&T source license.

Because this distribution tape requires a 4.2BSD license as well as an AT&T license, any 1984 Institutional or Supporting member wishing to receive the source version of the 84.2 distribution tape MUST send a copy of their 4.2BSD license to the USENIX Office for verification.

Also, because of the timeliness of this submission and its somewhat large size, the buglist will constitute the entire 84.2 Distribution Tape. Thus non-4.2 licensees would not benefit from this distribution and will not be receiving an 84.2 tape unless they specifically request it.

1984 Institutional and Supporting members must have a 1984 Tape Release Form on file at the Office before they may be sent a tape.

The USENIX Association wishes to thank mt Xinu for their efforts in preparing this submission and providing it to the UNIX community.
1984 Software Distribution Tape Submissions

Peter Gross
USENIX Tape Editor

The latest software distribution tape, the 84.1 tape, has been distributed to all 1984 Institutional and Supporting members who have a 1984 Tape Release Form and the necessary licenses on file. If you have not received your tape, contact the USENIX office.

Contents of the 84.1 USENIX Tape Submissions

The "restrictions" refer to AT&T licensing requirements. In the case of restricted distributions, versions were made with the restricted files excised if the remaining software was useful. Distributions with excised files will have a file named MISSING which lists the removed files.

Submitter: John Buck
Affiliation: New York Polytechnic Institute
Contents: 
  actng      accounting package for UNIX resource utilization.
  hash       hashed-password-file system. This is integrated into other programs included in this distribution, including login, ls, su and quotas.
  hasp       PWB UNIX hasp sub-system made to run under System V.
  include    Patches/additions to /usr/include (accounting, quotas, hashing header files, etc.).
  login      login program that does quota checking, hashed password file logins, etc.
  quotas     Quota subsystem.
  tron       A system security program.
  util        Various added/changed utilities.
  uts5       Modifications to the UNIX System V kernel.

Restrictions: System V, V7

Submitter: Charles LaBrec
Affiliation: Purdue University
Contents: Modifications to make(I) to allow it to work transparently with RCS (Revision Control System).

Restrictions: V7

Submitter: William Shannon
Affiliation: Sun Microsystems
Contents: Enhancements for 4.2BSD ARP (Address Resolution Protocol) code.
Restrictions: none
Release Form for 1985 USENIX Distribution Tapes

INSTITUTIONAL OR SUPPORTING MEMBERS ONLY

As an Institutional or Supporting Member of the USENIX Association, I wish to receive the 1985 USENIX Distribution Tape(s) (hereafter referred to as the TAPES). By virtue of this signed release form, I agree to the following Terms and Conditions governing the use of any and all TAPES sent to me.

1) Neither the USENIX Association, its officers, employees, and agents, nor the authors nor submitters of the materials on the TAPES (hereafter referred to as the ABOVENAMED) make any representation or warranty, expressed or implied, as to the accuracy, completeness, or usefulness, of the TAPES or the contents thereof for any purpose whatsoever.

2) To the extent permitted by law, I hereby explicitly indemnify and hold harmless the ABOVENAMED from any claims, costs, damages and liability whatsoever with respect to the TAPES or the contents thereof.

3) No property rights with respect to the TAPES or the contents thereof shall transfer to me. The TAPES shall remain the property of the USENIX Association and the contents thereof shall remain the property of the person or persons who possessed that property before the TAPES were written.

4) The licensing of my use of the materials on the TAPES by the USENIX Association is limited to those materials for which the USENIX Association is authorized to do so. The USENIX Association has attempted to verify that it has the right to distribute all of the materials on the TAPE, but it cannot guarantee that it has such right. In the event that any dispute arises as to the distributed materials, I agree that the USENIX Association may direct me to erase materials from the TAPES and that I will remove said materials from the TAPES, destroy all copies thereof, and make no claim against the ABOVENAMED for any action arising out of the use or the destruction of the materials.

5) I certify that I have the UNIX licenses or sub-licenses indicated below and that I shall use the materials on the TAPES so as not to violate said licenses. In particular, I agree that if there should be material on the TAPES that I am not licensed under our UNIX license(s) to receive or use, I shall neither use nor make copies of said materials.

6) I agree that this license grants me no rights to subdistribute any or all of the materials on the TAPES, except for use at the address designated below or on my membership form. This license grants me no rights to incorporate any of the materials on the TAPES in a commercial product, and I agree that, if such materials are not in the public domain, I will secure such rights from the owner of the property before incorporating any of the materials on the tapes in a commercial product.

In consideration of the mutual promises contained herein, I release and discharge the ABOVENAMED, their officers, employees and agents, of and from any claims, causes of action, costs or demands of whatever nature, whether anticipated or unanticipated, and whether known or unknown, which I have, may claim to have, or at any time heretofore have had against the above mentioned individuals and entities, or against any of them, including but not limited to, any and all claims, demands or causes of action which are contained in or may arise out of the circumstances set forth in this agreement, above.

The parties hereto acknowledge that they are familiar with the provisions of Section 1542 of the California Civil Code and expressly agree that the Release set forth above constitutes a waiver and release of any rights or benefits that they may have under California Civil Code Section 1542, which provides that:
"A general Release does not extend to claims which the creditor does not know or suspect to exist in his favor at the time of executing the release, which known by him must have materially affected his settlement with the debtor."

Subject to the above conditions, the USENIX Association grants and I hereby accept a **limited license to use** the materials on the TAPES at the address given below.

The contributions are generally separated into distribution tapes containing material covered by different source licenses. Tapes will include all material from earlier versions of source licenses which the current license specifically allows (i.e. the System III tape will include material licensed for V6, V7, etc. as well as no-disclosure material). **Binary license holders may receive ONLY the tape with no disclosure material.**

As an Institutional or Supporting Member, you are entitled to one tape per distribution. (However, additional different versions, for which you have appropriate source licenses, may be purchased. Contact the USENIX Office for more information.) Since there is no guarantee that material will be submitted for all license categories, please indicate the UNIX source licenses you own and your preference for receiving tapes (1 = first choice, 2 = second, etc.):

<table>
<thead>
<tr>
<th>Source License</th>
<th>Tape Preference</th>
</tr>
</thead>
<tbody>
<tr>
<td>System V</td>
<td>[ ]</td>
</tr>
<tr>
<td>System III</td>
<td>[ ]</td>
</tr>
<tr>
<td>UCB 4.x BSD</td>
<td>[ ]</td>
</tr>
<tr>
<td>Version/32V</td>
<td>[ ]</td>
</tr>
<tr>
<td>UCB 2.x BSD</td>
<td>[ ]</td>
</tr>
<tr>
<td>Version 7</td>
<td>[ ]</td>
</tr>
<tr>
<td>PWB/UNIX Version 1.0</td>
<td>[ ]</td>
</tr>
<tr>
<td>Phototypesetter</td>
<td>[ ]</td>
</tr>
<tr>
<td>Version 6</td>
<td>[ ]</td>
</tr>
<tr>
<td>Mini-UNIX</td>
<td>[ ]</td>
</tr>
<tr>
<td>UNIX/TSS</td>
<td>[ ]</td>
</tr>
<tr>
<td>UNIX/UNIXVAC</td>
<td>[ ]</td>
</tr>
<tr>
<td>Other:</td>
<td>[ ]</td>
</tr>
<tr>
<td>No-Disclosure tape</td>
<td></td>
</tr>
</tbody>
</table>

Tape Density (specify only if you cannot accept either density):

[ ] 800 bpi  [ ] 1600 bpi

Address to which your TAPES should be sent and where TAPES will be used, if different from address on membership form:
Institution/Member Name: ____________________________________________

Please Print

Member Number (if known): ____________

Authorized Signature: __________________________ Date: ____________

USENIX Association Signature: __________________________ Date: ____________

Please sign and return two copies of all three pages of this form to

USENIX Association Office
P.O. Box 7
El Cerrito, CA 94530

One copy will be signed by a USENIX representative and returned to you. Copies of your relevant UNIX license(s) must be on file at the USENIX Office for you to receive your tape. It is your responsibility to notify the USENIX Office should there be any change in your licensing categories, tape preference, or tape mailing address.

__________________________

For Office Use Only

date ____________  lic ____________  how ____________  by ______
type sent ____________  date ____________  by _____  db ____________

November 1984  Volume 9, Number 5
Application for Membership to the USENIX Association

Calendar Year 1985

Please type or print

[ ] New [ ] Renewal — Member number from mailing label: _________

Member Name: ____________________________________________

Address*: ________________________________________________

_________________________________________________________

Phone: ____________________________________________________

uucp network address: ______________________________________

Name and address of official Member Representative, if membership is for an institution:

Representative Name: _______________________________________

Address: _________________________________________________

_________________________________________________________

Phone: ____________________________________________________

uucp network address: ______________________________________

Individual, Institutional, and Supporting categories are all open to either institutions or individuals. Membership fees are:

[ ] $ 30 Individual  [ ] $ 15 Student (full-time)

[ ] $ 250 Institutional  [ ] $ 100 Institutional — educational discount
  Open to accredited educational institutions

[ ] $1000 Supporting

[ ] $ 9 Overseas airmail
  (Fee required for Individual or Student Memberships only)

[ ] Check enclosed: $ ___________  [ ] Purchase order enclosed; invoice required
  Payments must be in US dollars payable on a US bank

[ ] Check here if you do NOT want your name and address made available to other members, either verbally or in printed matter.

[ ] Check here if you do NOT want your name and address made available for commercial mailings.

* Accounting and billing address, if membership is for an institution

OVER
For Institutional and Supporting Memberships Only

Institutional and Supporting Members with UNIX source licenses may take advantage of certain services requiring license verification. Please indicate the type of source license(s) held:

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>__System V</td>
<td>__System III</td>
<td>__UCB 4.x BSD</td>
<td>__32V</td>
</tr>
<tr>
<td>__UCB 2.x BSD</td>
<td>__V7</td>
<td>__PWB/UNIX 1.0</td>
<td>__Phototypesetter</td>
</tr>
<tr>
<td>__Mini-UNIX</td>
<td>__UNIX/UNIVAC</td>
<td>__UNIX/TSS</td>
<td></td>
</tr>
<tr>
<td>__Other:</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

If you do hold UNIX source licenses, please send the signature page and the pages that show the version of UNIX you are licensed for, the name of the institution owning the license, and the type, serial number, and location of the CPUs. If you hold Berkeley licenses, send copies of those also. If you have more than one license, please send the above information for each. Institutional or Supporting members renewing their memberships need send only newly acquired licenses or those not previously verified by USENIX.

Institutional and Supporting Members (whether new or renewing), with or without source licenses, must also complete a **1985 Tape Release Form** if they wish to receive distribution tapes.

- [ ] Licenses(s) enclosed
- [ ] Licenses(s) already on file
- [ ] Tape Release enclosed

Authorized Signature: ________________________________ Date: ____________

Please complete and return this form to:

USENIX Association  
P. O. Box 7  
El Cerrito, CA  94530

This form **must** accompany your purchase order or payment. You will receive a card acknowledging your membership as soon as it is processed.
Announcing Second Printing of USENIX 4.2BSD Manuals

The second USENIX sponsored printing of 4.2BSD manuals is in progress and deliveries can be expected 6 - 8 weeks from now. The second printing will be much the same as the previous, although a couple of pages from the User's Manual that were missing on the first printing have been added, a color coded binding comb has also been added to make identification of individual volumes easier, and a couple of other minor corrections have been made. Prices remain unchanged. An order form is included elsewhere in this newsletter for those Institutional and Supporting Members holding a 4.2BSD source license that wish to purchase manuals.

For the convenience of those who have manuals from the first printing, the pages missing from the User's Manual Reference Guide are reproduced below.
4.2BSD Manual Reproduction Authorization and Order Form

Please return both copies

USENIX Member No.:underline
Purchase Order No.:underline
Date:underline

As the representative of a USENIX Association Institutional or Supporting Member in good standing, and as a bona fide license holder of both a 4.2BSD software license from the Regents of the University of California and a UNIXtm/32V or System III or System V license, or sublicense from Western Electric Company, and pursuant to the copyright notice as found on the rear of the cover page of the UNIX/32V Programmer’s Manual stating that

"Holders of a UNIXtm/32V software license are permitted to copy this document, or any portion of it, as necessary for licensed use of the software, provided this copyright notice and statement of permission are included."

I hereby appoint the USENIX Association as my agent, to act on my behalf to duplicate and provide me with such copies of the Berkeley 4.2BSD Manuals as I may request.

Signed:underline
Institution:underline

Ship to:underline Billing address, if different:
Name:underline Name:underline

Phone:underline Phone:underline

The prices below do not include shipping and handling charges or state or local taxes. All payments must be in US dollars drawn on a US bank.

4.2BSD User’s Manual at $18.50 each = $underline
4.2BSD Programmer’s Manual at $17.00 each = $underline
4.2BSD System Manager’s Manual at $ 9.50 each = $underline
Total $underline

[ ] Purchase order enclosed; invoice required
(Purchase orders must be enclosed with this order form.)

[ ] Check enclosed for the manuals: $underline
(Howard Press will send an invoice for the shipping and handling charges and applicable taxes.)

Make your check or purchase order out to Howard Press and mail it with this order form to:
Howard Press
c/o USENIX Association
P.O. Box 7
El Cerrito, CA 94530

for office use. l.v.;underline check no.:underline amt. rec’d:underline
USENIX Software Distribution Tapes

4.2BSD Buglist

Winter '85 Meeting